UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Paul Reuben Day and Brian Robert Muras

Appeal No. _____
Application No. 10/754,011

APPEAL BRIEF

Attorney Docket No. ROC920030217US1 PATENT
Confirmation No. 7133

CERTIFICATE OF ELECTRONIC TRANSMISSION
I hereby certify that this correspondence for Application No. 10/754,011 is being electronically transmitted to Technology Center 2167 via EFS-WEB, on September 28, 2009.

/Scott A. Stinebruner/ September 28, 2009
Scott A. Stinebruner, Reg. No. 38,323 Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Paul Reuben Day, et al.	Art Unit:	2167
Application No.:	10/754,011	Examiner:	Kimberly M. Lovel
Filed:	January 8, 2004		
For:	METHOD APPLYING TRANSITIVE CLOSURE TO GROUP BY AND ORDER BY CLAUSES		

Mail Stop Appeal Brief - Patents
Commissioner for Patent
P.O. Box 1450
Alexandria, VA 22213-1450

APPEAL BRIEF

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation, of Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1, 3-5, 7-10, 13, 15-19, and 25-29 are pending in the Application, stand rejected, and are now on appeal. Claims 2, 6, 11-12, 14 and 20-24 have been canceled.

IV. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection mailed April 27, 2009.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant's invention is generally directed to a database engine and optimizer framework that support the use of transitive closure to assist in rewriting GROUP BY and ORDER BY clauses to reduce the number of referenced tables (optimally to a single table if possible) and/or to free the join order selected for the query plan. (Application, paragraph [0031]).

A GROUP BY clause aggregates records in the result set that have a common value in a specified field or fields. An ORDER BY clause arranges the records of the result set in a specified order. Both clauses are therefore used to re-order a result set that is returned for a query. (Application, paragraph [0009]).

GROUP BY and ORDER BY clauses have been found to complicate query optimization in some instances, as the need to re-order a result set to comply with a GROUP BY or ORDER BY clause often requires the creation of temporary data structures and/or restricts the ability of an optimizer to test all possible optimizations. For example, when a GROUP BY or ORDER BY clause includes references to more than one table, the optimizer often must create a temporary file to hold the result set in order to perform the GROUP BY or ORDER BY operation. The creation of the temporary file, however, often slows the performance of the query. (Application, paragraph [0026]). In contrast, when a GROUP BY or ORDER BY clause references only one table, creation of a temporary file is not needed in order to re-order a result set, so the aforementioned performance penalties can be avoided. (Application, paragraph [0028]).

Likewise, when an ORDER BY or GROUP BY clause is present in a join operation, the optimizer is locked into a particular join order even if that order may not be optimal when performing the ordering or grouping according to an index. This requirement prevents the optimizer from selecting a join order that may be more optimal. (Application, paragraph [0030]).

Embodiments of the invention address the aforementioned problems through the use of transitive closure to assist in rewriting GROUP BY and ORDER BY clauses to reduce the

number of referenced tables (optimally to a single table if possible) and to free the join order selected for the query plan. By performing transitive closure on the selection, or search, conditions, a query optimizer is able to identify which fields referenced by the ORDER BY and GROUP BY clauses can be replaced with equivalent fields to improve the performance of the query. (Application, paragraph [0038]).

Transitive closure is a technique useful with directed graphs that also has applicability to SQL optimization. Fundamentally, transitive closure determines that if A=B and B=C, then A=C. (Application, paragraph [0038]).

For example, transitive closure analysis may be applied to a query such that the criteria may be re-written in such a manner as to reduce the number of tables referenced thereby to a single table, which may also provide the benefit of eliminating the need to create a temporary table as might otherwise be required. (Application, paragraphs [0032]-[0033]). Transitive closure analysis may also be applied to a plurality of search conditions in a query to determine a subset of equivalent search fields, and based thereupon, rewrite the query to re-order a result set of the database query to generate a set of respective modified criteria that each reference one or more equivalent search fields, such that a join order may be selected from among a plurality of join orders. (Application, paragraphs [0031]-[0034]).

For the convenience of the Board, claims 1, 10, 13, 25, 27, and 29, the independent claims, have been reproduced below and annotated with references to the specification and drawings to satisfy the requirement to concisely explain the claimed subject matter:

Independent Claim 1

A method (Application, Fig. 3, paragraph [0064]) for optimizing a database query (Application, Fig. 2, block 46) in a computer of the type including a database management system (Application, Fig. 1, blocks 10, 32), the database query including criteria that references a plurality of tables in order to re-order a result set generated for the database query (Application, paragraphs [0052]-[0056]), wherein the criteria is one of a GROUP BY clause and an ORDER BY clause (Application, paragraph [0064]), the method comprising the steps of, in the computer:

> *applying transitive closure analysis to at least one search condition in the query to identify an equivalent field for a field referenced in the criteria (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]); and*

based on the transitive closure analysis, rewriting the criteria to generate modified criteria to reduce the number of tables referenced thereby by substituting the equivalent field for the field referenced in the criteria, including rewriting the criteria to generate modified criteria that references only one table, based on the transitive closure analysis (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]).

Independent Claim 10

A method (Application, Fig. 3, paragraph [0064]) of optimizing a database query (Application, Fig. 2, block 46) in a computer of the type including a database management system (Application, Fig. 1, blocks 10, 32), the database query including criteria that operates to re-order a result set of the database query and requires creating a temporary file during operation (Application, paragraphs [0052]-[0056]), wherein the criteria is one of a GROUP BY clause and an ORDER BY clause (Application, paragraph [0064]), the method comprising the steps of, in the computer:

applying transitive closure analysis to at least one search condition in the query to identify an equivalent field for a field referenced in the criteria (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]); and

rewriting the criteria, based on the transitive closure analysis, to generate a modified criteria by substituting the equivalent field for the field referenced in the criteria, wherein the criteria references a plurality of tables and the modified criteria references a single table; said modified criteria operating to re-order a result set of the database query and avoid creating a temporary file during operation (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]).

Independent Claim 13

A method (Application, Fig. 3, paragraph [0064]) for optimizing a database query (Application, Fig. 2, block 46) in a computer of the type including a database management system (Application, Fig. 1, blocks 10, 32), the database query involving a plurality of join operations and a plurality of search conditions (Application, paragraph [0069]-[0075]), the method comprising the steps of, in the computer:

applying transitive closure analysis to the plurality of search conditions in the query to determine a subset of equivalent search fields (Application, Fig. 3, blocks 302-312, paragraphs [0064]-[0068] and [0091]);

rewriting a criteria, that operates to re-order a result set of the database query, to generate a set of respective modified criteria that each reference one or more equivalent search fields, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause (Application, Fig. 3, blocks 302-312, paragraphs [0064]-[0068] and [0091]); and

selecting a join order from among a plurality of join orders for the plurality of join operations, including analyzing join orders using at least one of the set of respective

modified criteria (Application, Fig. 3, blocks 302-312, paragraphs [0064]-[0068] and [0091]).

Independent Claim 25

A program product (Application, paragraph [0047]), comprising:

program code (Application, paragraph [0047]) configured upon execution thereof to:

apply transitive closure analysis to at least one search condition in a query that includes criteria that references a plurality of tables in order to re-order a result set generated for the query (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]), and based on the transitive closure analysis, rewrite the criteria to generate modified criteria to reduce the number of tables referenced thereby (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]), wherein the criteria is one of a GROUP BY clause and an ORDER BY clause (Application, paragraph [0064]), wherein the program code is configured to apply transitive closure analysis to identify from the at least one search condition an equivalent field for a field referenced in the criteria, and wherein the program code is configured to rewrite the criteria by substituting the equivalent field for the field referenced in the criteria (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]); and

a recordable computer readable medium (Application, paragraph [0047]) storing the program code.

Independent Claim 27

A program product (Application, paragraph [0047]), comprising:

program code (Application, paragraph [0047]) configured upon execution to:

apply transitive closure analysis to a plurality of search conditions to determine a subset of equivalent search fields within a database query involving a plurality of join operations and the plurality of search conditions (Application, Fig. 3, blocks 302-312, paragraphs [0064]-[0068] and [0091]), rewrite a criteria, that operates to re-order a result set of the database query, to generate a set of respective modified criteria that each reference one or more equivalent search fields (Application, Fig. 3, blocks 302-312, paragraphs [0064]-[0068] and [0091]), and select a join order from among a plurality of join orders for the plurality of join operations by analyzing join orders using at least one of the set of respective modified criteria (Application, Fig. 3, blocks 302-312, paragraphs [0064]-[0068] and [0091]), wherein the criteria is one of a GROUP BY clause and an ORDER BY clause (Application, paragraph [0064]); and

a recordable computer readable medium (Application, paragraph [0047]) storing the program code.

Independent Claim 29

An apparatus (Application, Fig. 1, block 10), comprising:

at least one processor (Application, Fig. 1, block 12);

a memory coupled (Application, Fig. 1, block 14) with the at least one processor; and

a program code (Application, paragraph [0047]) residing in memory and executed by the at least one processor, the program code configured to apply transitive closure analysis to at least one search condition in a query that includes criteria that references a plurality of tables in order to re-order a result set generated for the query (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]), and based on the transitive closure analysis, rewrite the criteria to generate modified criteria to reduce the number of tables referenced thereby (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]), wherein the criteria is one of a GROUP BY clause and an ORDER BY clause (Application, paragraph [0064]), wherein the program code is configured to apply transitive closure analysis to identify from the at least one search condition an equivalent field for a field referenced in the criteria, and wherein the program code is configured to rewrite the criteria by substituting the equivalent field for the field referenced in the criteria (Application, Fig. 3, blocks 302-308, paragraphs [0064]-[0068]).

Other support for the claimed subject matter may generally be found in paragraphs [0052]-[0063], [0066], [0069]-[0080] and [0083]-[0090]. In addition, it should be noted that, as none of the claims recite any means plus function or step plus function elements, no identification of such elements is required pursuant to 37 CFR §41.37(c)(1)(v). Furthermore, there is no requirement in 37 CFR §41.37(c)(1)(v) to provide support for the subject matter in the separately argued dependent claims, as none of these claims recite means plus function or step plus function elements, and so no discussion of any of these claims is provided.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1, 3-5, 9, 10, 13, 15, 18, 19, and 25-29 are rejected under 35 U.S.C. § 103 (a) as being unpatentable over Pham et al. (US Patent No. 6,757,677) (hereinafter "*Pham*") in view of Ghazal et al. (US Patent No. 6,662,175) (hereinafter

"*Ghazal*") and in view of Bulletin of the Technical Committee on Data Engineering (hereinafter "*Data Engineering*").

B. Claims 7-8 and 16-17 are rejected to under 35 U.S.C. § 103 (a) as being unpatentable over *Pham*, *Ghazal* and *Data Engineering*, and in further view of Chaudhuri (US Patent No. 5,598,559) (hereinafter "*Chaudhuri*").

VII. ARGUMENT

Applicant respectfully submits that the Examiner's rejections of claims 1, 3-5, 7-9, 10, 13, 15-19 and 25-29 are not supported on the record, and should be reversed. All such claims have been rejected as being obvious over the prior art cited by the Examiner. Appellant respectfully submits that, in the instant case, the Examiner has failed to establish a *prima facie* case of obviousness as to the aforementioned claims, and thus, the rejections thereof should be reversed.

Based on the Supreme Court's decision in KSR International Co. v. Teleflex Inc., 127 S. Ct. 1727, 1734, 82 USPQ2d 1385, 1382 (2007), a *prima facie* showing of obviousness still requires that the Examiner establish that the differences between a claimed invention and the prior art "are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art." 35 U.S.C. §103(a). Such a showing requires that all claimed features be disclosed or suggested by the prior art. Four factors generally control an obviousness inquiry: 1) the scope and content of the prior art; 2) the differences between the prior art and the claims; 3) the level of ordinary skill in the pertinent art; and 4) secondary considerations of non-obviousness, such as commercial success of products covered by the patent claims, a long felt but unresolved need for the invention, and failed attempts by others to make the invention. KSR, 127 S. Ct. at 1734 (quoting Graham v. John Deere Company, 383 U.S. 1, 17-18 (1966)) ("While the sequence of these questions might be reordered in any particular case, the [Graham] factors continue to define the inquiry that controls."). Moreover, in KSR, the Court explained that "[o]ften, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue" and "[t]o facilitate

review, this analysis should be made explicit." KSR, 127 S. Ct. at 1740-41 citing In re Kahn, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."). But, not every combination is obvious "because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known." KSR, 127 S. Ct. at 1741.

As a result, after KSR, while there is no rigid requirement for an explicit "teaching, suggestion or motivation" to combine references, there still must be some evidence of "a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does" in an obviousness determination. KSR, 127 S. Ct. at 1731.

Appellant respectfully submits that, in the instant case, the Examiner has failed to establish a *prima facie* case of obviousness as to the currently rejected claims, and as such, the rejections thereof should be reversed. Applicant's remarks in rebuttal to the Examiner's rejections are presented below, starting with the relevant independent claims and followed up by a discussion of selected dependent claims. In some cases, specific discussions of particular claims are not made in the interest of streamlining the appeal. The omission of a discussion with respect to any particular claim, however, should not be interpreted as an acquiescence as to the merits of the Examiner's rejection of the claim, particularly with respect to claims reciting features that are addressed in connection with the rejection applied to other claims pending in the appeal.

A. Claims 1, 3-5, 9, 10, 13, 15, 18, 19, and 25-29 are patentable over *Pham* and *Ghazal*, and in view of *Data Engineering*.

Independent Claim 1

Independent claim 1 recites a method for optimizing a database query in a computer of the type including a database management system, where the database query includes criteria that references a plurality of tables in order to re-order a result set generated for the database query, and where the criteria is one of a GROUP BY clause and an ORDER BY clause. The claimed method includes the steps of applying transitive closure analysis to at least one search

condition in the query to identify an equivalent field for a field referenced in the criteria, and based on the transitive closure analysis, rewriting the criteria to generate modified criteria to reduce the number of tables referenced thereby by substituting the equivalent field for the field referenced in the criteria, including rewriting the criteria to generate modified criteria that references only one table, based on the transitive closure analysis.

It is important to note, therefore, that claim 1 is directed in part to the concept of using transitive closure analysis to reduce the number of tables referenced by a GROUP BY or ORDER BY clause in a query, which originally references multiple tables, down to a single table.

In rejecting claim 1, the Examiner is required to rely on a total of three separate references, *Pham*, *Ghazal* and *Data Engineering*. Applicant respectfully submits, however, that the proposed combination fails to disclose or suggest each and every feature of claim 1, and therefore, the rejection should be reversed.

The primary reference to *Pham* is generally directed to performing a join of multiple tables in response to receiving a query containing WHERE and GROUP BY clauses. The join is performed by reducing the number of active rows of at least one of the tables to be joined prior to performing the join operation. The Examiner admits on page 4 of the Final Office Action that while *Pham* discloses a query with at least one search condition (WHERE clause), *Pham* fails to disclose the further limitations of applying transitive closure analysis to at least one search condition in the query, rewriting the criteria to generate modified criteria to reduce the number of tables being referenced, and based on the transitive closure analysis, substituting the equivalent field for the field referenced in the criteria to generate modified criteria that references only one table.

The Examiner contends that these deficiencies are supplied by *Ghazal*, contending that *Ghazal* discloses at col. 1, lines 7-9 and 22-26 query optimization applying transitive closure to a search condition in the query to identify an equivalent field for the field referenced in the criterion. The Examiner also contends that *Ghazal* discloses that based on the transitive closure analysis, the criteria is rewritten to generate modified criteria to reduce the number of tables referenced by substituting the equivalent field for the reference field in the criteria to generate modified criteria at col. 1, lines 37-38.

Ghazal contains two occurrences of the term transitive closure; however, neither is particularly relevant to the specific usage of transitive closure analysis as is recited in claim 1. The first occurrence is in the passage cited by the Examiner in the background of the disclosure. Here *Ghazal* merely states that transitive closure is one of a number of syntactic or algebraic transformations that may be used for query transformation. However this passage gives no further details to how one of ordinary skill in the art would use any of the disclosed syntactic or algebraic transformations to assist the query optimizer in performing query transformations. This passage certainly fails to disclose or even suggest applying transitive closure analysis to at least one search condition in the query to identify an equivalent field referenced in the criteria and using the analysis to rewrite the criteria to reduce the number of tables referenced thereby, as is required by claim 1.

The second occurrence of the term transitive closure can be found in *Ghazal* at col. 5 line 57- col. 6 line 8. This passage discusses performing a transitive closure of the WHERE-clause conditions to calculate new date constraints which could reduce the size of the intermediate result that is generated. This passage also fails to disclose or suggest using transitive closure analysis to rewrite the criteria to reduce the number of tables referenced in a query. Moreover, *Ghazal* fails to disclose or suggest "applying transitive closure analysis to at least one search condition in the query [having a GROUP BY or ORDER BY clause] to identify an equivalent field for a field reference in the criteria" and "based on the transitive closure analysis, rewriting the criteria to generate modified criteria to reduce the number of tables referenced thereby," as required by claim 1.

On page 4 of the Final Office Action, the Examiner also states that it would have been obvious to one of ordinary skill in the art to apply the transitive closure disclosed in *Ghazal* to optimize the query of *Pham*. The Examiner states that one would have been motivated to do so since it is well known that query optimization improves overall performance which reduces resource utilization. However, even if the transitive closure techniques disclosed in *Ghazal* with respect to WHERE clauses were applied to the GROUP BY operation of *Pham*, one of ordinary skill in the art would use the transitive closure analysis to assist in reducing the number of rows prior to a join operation as disclosed in *Pham*. It is only through hindsight and the benefit of Applicant's disclosure that the Examiner is able to assert that one skilled in the art would use the

transitive closure analysis in *Ghazal* in *Pham* to "rewrit[e] the criteria to generate modified criteria to reduce the number of tables referenced thereby by substituting the equivalent field for the field referenced in the criteria" as recited in Applicant's claim 1, when there is no teaching or suggestion in either reference to reduce the number of tables. Rather both references teach reducing table sizes (i.e., number of rows) prior to a join operation.

On page 4 of the Final Office Action, and with respect to the further recitation in claim 1 directed to rewriting the criteria to generate modified criteria that references only one table, the Examiner also asserts that *Ghazal* discloses referencing only one row at col. 1, lines 27-28, but admits that the reference fails to disclose referencing only one table. The Examiner then makes a conclusory statement that it would have been obvious to apply the concept of referencing only one row in order to reference only one table. However, one of ordinary skill in the art would know that there is a substantial difference between referencing rows of table and referencing tables generally. Moreover the Examiner has failed to provide any rationale for this statement except that "[o]ne would have been motivated to do so since this is the basic purpose of query rewrite." Yet, the Examiner has failed to show any recognized link in the art between rewriting a query and reducing the number of tables referenced by a query to reference only one table. One of ordinary skill in the art would recognize that query optimizers frequently rewrite queries in order to reduce the time and resources required to perform the query without necessarily reducing multi-table queries to a single table. It is only with hindsight and the benefit of Applicant's disclosure that the Examiner could assert this motivation.

Perhaps recognizing the weakness of such an assertion, the Examiner is forced to rely on a third reference, *Data Engineering*, admitting that the combination of *Pham* and *Ghazal* fails to explicitly disclose that the transitive closure reduced the number of tables referenced. *Data Engineering* generally discusses the evolution of query optimization with a section 3 on pages 6-8 briefly discussing query transformation. This section generally discusses transformations such as distributing NOT predicates where appropriate, bounding LIKE predicates, converting disjuncts to conjuncts with DeMorgan's law, and in the section cited by the Examiner, generating additional equality predicates using transitive closure of equality predicates.

The Examiner contends that *Data Engineering* discloses the concept of using transitive closure on both single-table and join predicates, including applying transitive closure to reduce

the number of tables referenced on page 7, lines 17-23. The Examiner's interpretation of this passage, however, is flawed. The passage referred to by the Examiner begins by stating that, "DB2 for MVS generates the transitive closure of equality predicates, for both single-table and join predicates, to allow earlier filtration of rows and more potential join sequences." (emphasis added). This passage fails to disclose reducing the number of tables.

Further, the Examiner points to the join example, which illustrates that "join predicates of T1.C1=T2.C2 AND T2.C2=T3.C3 will cause DB2 to generate T1.C1=T3.C3." The Examiner leaps to the conclusion that this is a reduction of the number of tables even though the remainder of the statement points out that "without [generating the T1.C1=T3.C3 predicate] the join enumerator would have deferred considering a join between T1 and T3." This passage illustrates that with the use of transitive closure, the join of the T1 and T3 tables can be performed earlier than if no transitive closure were performed. Therefore, *Data Engineering* also fails to disclose "rewriting the criteria to generate modified criteria to reduce the number of tables referenced thereby" as recited in Applicant's claim 1. Rather, *Data Engineering* discloses generation of additional predicates to "allow earlier filtration of rows and more potential join sequences."

Further, the passage concludes that "[w]hen there are more than a preset number of tables in a join, join predicate transitive closure is not performed in order to keep the search space of the dynamic programming join enumeration strategy in check." (emphasis added). In addition to failing to disclose using transitive closure to reduce the number of tables, *Data Engineering* further discourages using transitive closure when a preset number of tables is exceeded. *Data Engineering* therefore adds little to the rejection.

In addition, it is notable that none of the references disclose transitive closure being used with a GROUP BY or an ORDER BY clause. *Pham* discloses a GROUP BY statement but fails to disclose transitive closure. *Ghazal* uses transitive closure with a WHERE clause, not a GROUP BY or ORDER BY clause. *Data Engineering* uses transitive closure with a JOIN clause. Thus *Ghazal* and *Data Engineering* do not consider some of the challenges of using GROUP BY and ORDER BY clauses as set forth in the background of Applicant's specification. Applicant submits that it is only through hindsight and the benefit of Applicant's disclosure that the Examiner could assert using the transitive closure techniques in *Ghazal* and/or *Data Engineering* (which have only been used with JOIN or WHERE clauses) with the GROUP BY

clause in *Pham*. Moreover the Examiner has provided no additional information or citations in the disclosures directed to how transitive closure could be used with a GROUP BY or ORDER BY clause. The Examiner merely provides a conclusory statement based on the combination.

The Examiner additionally makes several arguments in rebuttal on pages 15-17 of the Final Office Action. First, the Examiner argues that there is no appreciable difference between referencing rows of a table or referencing different tables since the concept of dividing a database into a plurality of tables is known in the art. Applicant strongly disagrees with this assertion, since there are well recognized instances where referencing multiple tables vs. a single table has a notable effect on how a query is optimized and executed, which does not apply to referencing multiple rows vs. a single row. In fact, one such instance is described in paragraphs [0026]-[0027] of the Application, where it is discussed that when a GROUP BY or ORDER BY clause references more than one table, a temporary table must be created during execution, which temporary table is not required when a GROUP BY or ORDER BY clause references only a single table. The passage also notes that in the case of an interactive query, this can lead to unresponsiveness due to the need to complete execution of a query before returning results, which would otherwise not be required were only a single table referenced. There is simply no analogous situation that would arise based upon whether one row is referenced verses multiple rows, and Applicant therefore submits that the Examiner's attempt to analogize rows to tables in this context is misplaced.

Second, the Examiner apparently argues that since *Ghazal* discloses the use of transitive closure to rewrite queries, and that since the goal of doing so reduces an expression to the smallest possible relational space, apparently any use of transitive closure analysis in connection with query optimization is apparently obvious. Applicant submits that the Examiner's argument in this respect is entirely too broad, because the Examiner has failed to even find any recognition in the art of the desirability of applying transitive closure analysis to GROUP BY or ORDER BY clauses, much less to do so for the purpose of reducing the number of tables referenced by such a clause down to only one table.

Third, the Examiner argues that transitive closure is used in *Data Engineering* to reduce the number of tables being referenced. Irrespective of whether or not this is the case, this passage still fails to disclose or suggest reducing the number of tables being referenced in a

GROUP BY or ORDER BY clause, or to reduce the number of tables to only one table, so the Examiner has still failed to establish a *prima facie* case of obviousness as to claim 1.

Fourth, the Examiner argues that since GROUP BY and ORDER BY clauses are merely specific types of "search criteria" it would have been obvious to use transitive closure analysis. It should be noted, however, that irrespective of whether this statement is true, GROUP BY and ORDER BY clauses do not serve the purpose of restricting the returned results, but instead serve to organize the results in a particular manner (grouped or ordered based upon a criteria). The examples in the art of transitive closure cited by the Examiner may very well apply to search criteria that restrict the results that are returned (e.g., the WHERE clause to which transitive closure is applied in *Ghazal*), yet none are applied to a criteria that is used to group or order search results as in the case of a GROUP BY or ORDER BY clause, which is fundamentally different therefrom. Applicant submits that again the Examiner is taking an overly broad interpretation of what would be obvious to one of ordinary skill in the art. Applicant's specification has served as a blueprint for the Examiner's rejection, and the Examiner has relied heavily on hindsight in making the rejection. Were hindsight properly disregarded, one would recognize that the art cited by the Examiner fails to appreciate the desirability of applying transitive closure analysis to a GROUP BY or ORDER BY clause for the purpose of reducing the number of tables referenced thereby.

Therefore, Applicant submits that the Examiner has failed to establish a *prima facie* case of obviousness and Applicant's independent claim 1, as none of the references, alone or in combination, discloses or suggests at least the use of transitive closure analysis to reduce the number of tables referenced by a GROUP BY or ORDER BY clause in a query down to only one table. Claim 1 is therefore patentable over the combination of *Pham*, *Ghazal*, and *Data Engineering*. Reversal of the Examiner's rejection of claim 1 is therefore respectfully requested.

Independent Claim 10

Claim 10 generally recites a method of optimizing a database query in a computer of the type including a database management system, where the database query includes criteria that operates to re-order a result set of the database query and requires creating a temporary file during operation, and where the criteria is one of a GROUP BY clause and an ORDER BY

clause. The method includes applying transitive closure analysis to at least one search condition in the query to identify an equivalent field for a field referenced in the criteria, and rewriting the criteria, based on the transitive closure analysis, to generate a modified criteria by substituting the equivalent field for the field referenced in the criteria. The criteria references a plurality of tables and the modified criteria references a single table, and the modified criteria operates to re-order a result set of the database query and avoid creating a temporary file during operation.

The Examiner has also rejected independent claim 10 as being unpatentable over *Pham*, *Ghazal*, and *Data Engineering*. Similar to the rejection of claim 1, the Examiner states that *Pham* fails to disclose generating a modified criteria using transitive closure analysis by substituting the equivalent field for the field references in the criteria, where the criteria references a plurality of tables and the modified criteria references a single table. The Examiner then admits on page 7 of the Final Office Action that "while *Ghazal* discloses referencing only one row, *Ghazal* fails to disclose referencing only one table." The Examiner further states that it would have been obvious to one of ordinary skill in the art to apply the concept of referencing only one row in order to reference only one table and one would be motivated to do so since this is the basic purpose of query rewrite. However, as noted above in connection with the rejection of claim 1, one of ordinary skill in the art would recognize that there is a substantial difference between referencing rows of table and referencing tables generally and it is only with the use of hindsight that the Examiner could make such an assertion.

Moreover, as set forth above, neither *Pham* nor *Ghazal* disclose or suggest using transitive closure analysis to rewrite query criteria that references multiple tables to reference a single table and that it is only through hindsight and the benefit of Applicant's disclosure that the Examiner can make these assertions. The Examiner then similarly asserts that *Data Engineering* discloses the missing elements. But as set forth above, *Data Engineering* also fails to disclose rewriting query criteria that references multiple tables to reference a single table.

Therefore, for the same or similar reasons as set forth with respect to independent claim 1, independent claim 10 is also patentable over *Pham*, *Ghazal*, and *Data Engineering*.

In addition, it should also be noted that claim 10 further recites that the database query includes criteria that "requires creating a temporary file during operation," and that rewriting the query "avoid[s] creating a temporary file during operation." Notably, the Examiner does not

even attempt to address this language in the rejection beyond simply parroting back the claim language. None of the art of record discloses or suggests any rewrite of a query to avoid the creation of a temporary file, nor could the Examiner assert as such. Based upon this reason alone, the Examiner has failed to establish a *prima facie* case of obviousness as to claim 10.

Applicant therefore respectfully submits that claim 10 is non-obvious over the prior art cited by the Examiner. Reversal of the Examiner's rejection of claim 10 is therefore respectfully requested.

Independent Claim 13

Claim 13 generally recites a method for optimizing a database query in a computer of the type including a database management system, where the database query involves a plurality of join operations and a plurality of search conditions. The method includes applying transitive closure analysis to the plurality of search conditions in the query to determine a subset of equivalent search fields, rewriting a criteria, that operates to re-order a result set of the database query, to generate a set of respective modified criteria that each reference one or more equivalent search fields, where the criteria is one of a GROUP BY clause and an ORDER BY clause, and selecting a join order from among a plurality of join orders for the plurality of join operations, including analyzing join orders using at least one of the set of respective modified criteria.

As noted above, and as is discussed in paragraph [0030] of the Application, conventionally, when an GROUP BY or ORDER BY clause is present in a query, an optimizer may be locked into a particular join order even if that order may not be optimal when performing the ordering or grouping according to an index. Specifically, as discussed in paragraph [0068] of the Application, the presence of a GROUP BY or ORDER BY clause that is implemented by an index limits the possible join orders from which the optimizer can select, and typically requires that the first table in the join order be the table over which the index is built. However, a more optimal join order may be obtained when the table over which an index is built is not first in the join order. Thus, claim 13 is directed in part to performing transitive closure analysis to effectively free an optimizer to select an optimal join order in a query that includes a GROUP BY or ORDER BY clause.

The Examiner has also rejected independent claim 13 as being unpatentable over *Pham*, *Ghazal*, and *Data Engineering*. The Examiner states on pages 9-10 of the Final Office Action that *Ghazal* discloses query optimization including the further limitations of applying transitive closure analysis to a plurality of search conditions in the query to determine a subset of equivalent fields and rewriting a criteria to generate a set of respective modified criteria that each reference one more equivalent search fields in col. 1 lines 7-9 and 22-36, which the Examiner admits is not disclosed in *Pham*. The Examiner additionally states that *Ghazal* discloses selecting join order from among a plurality of join orders for the plurality of join operations using at least one of the set of respective modified criteria at col. 1, lines 37-38, which the Examiner admits is also not disclosed in *Pham*.

As set forth above with respect to claim 1, col. 1 of *Ghazal* fails to disclose applying transitive closure to a plurality of search conditions. The background of *Ghazal* merely lists transitive closure as one of a number of techniques used by query optimizers with no further teachings of how they are used. Further, lines 37-38 of *Ghazal* disclose, "[t]he basic purpose of a query rewrite is to reduce the number of rows processed during the query." (*emphasis added*). Nowhere in this passage does *Ghazal* disclose selecting join order from among a plurality of join operations as contended by the Examiner. Furthermore as set for the above, *Ghazal* discloses using transitive closure analysis to reduce the size of an intermediate result at col. 5, line 56 – col. 6 line 17. However, *Ghazal* fails to disclose applying the analysis to a plurality of search conditions, much less selecting a join order from among a plurality of join orders for the plurality of join operations using at least one of the set of respective criteria modified by the transitive closure analysis.

Moreover, the Examiner admits that *Pham* and *Ghazal* fail to explicitly disclose transitive closure being used to reduce the number of tables and as above asserts that *Data Engineering* remedies the deficiency. However, claim 13 does not even recite reducing the number of tables referenced in a query, so Applicant fails to see how or why the Examiner cites *Data Engineering* in this context.

Applicant submits that none of the references cited by the Examiner, alone or in combination, discloses or suggests the concept of using transitive closure analysis on a GROUP BY or ORDER BY clause to generate modified criteria, and thereafter analyzing join orders

using at least one of the set of respective modified criteria. Therefore the combination of *Pham*, *Ghazal*, and *Data Engineering* fails to disclose all of the elements of Applicant's claim 13. Furthermore, the Examiner has provided no motivation to modify the combination of *Pham*, *Ghazal*, and *Data Engineering* to contain the elements of Applicant's claim 13. The Examiner merely makes a generalized statement that query optimization improves overall performance which reduces resource utilization, which falls far short of the motivation required to establish a *prima facie* case of obviousness as to claim 13. For these reasons, Applicant submits that independent claim 13 is patentable over *Pham*, *Ghazal*, and *Data Engineering* and respectfully requests that the rejection for claim 13 be withdrawn.

Independent Claim 25

Claim 25 generally recites a program product including in part program code configured to apply transitive closure analysis to at least one search condition in a query that includes criteria that references a plurality of tables in order to re-order a result set generated for the query, and based on the transitive closure analysis, rewrite the criteria to generate modified criteria to reduce the number of tables referenced thereby. The criteria is one of a GROUP BY clause and an ORDER BY clause, and the program code is configured to apply transitive closure analysis to identify from the at least one search condition an equivalent field for a field referenced in the criteria, and to rewrite the criteria by substituting the equivalent field for the field referenced in the criteria.

As discussed above in connection with claim 1, none of the references cited by the Examiner disclose or suggest applying transitive closure analysis to a GROUP BY or ORDER BY clause for the purpose of reducing the number of tables referenced thereby. In addition, there is a fundamental difference between reducing the number of rows referenced by a query and reducing the number of tables, so the Examiner's attempt to find analogy between the two is improper and is inconsistent with the understanding of those of skill in the art. Accordingly, claim 25 is patentable for many of the same reasons as set forth above for claim 1, and reversal of the rejection of claim 25 is respectfully requested.

Independent Claim 27

Claim 27 generally recites a program product including in part program code configured to apply transitive closure analysis to a plurality of search conditions to determine a subset of equivalent search fields within a database query involving a plurality of join operations and the plurality of search conditions, and rewrite a criteria, that operates to re-order a result set of the database query, to generate a set of respective modified criteria that each reference one or more equivalent search fields, and select a join order from among a plurality of join orders for the plurality of join operations by analyzing join orders using at least one of the set of respective modified criteria, where the criteria is one of a GROUP BY clause and an ORDER BY clause.

As discussed above in connection with claim 13, none of the references cited by the Examiner disclose or suggest applying transitive closure analysis to a GROUP BY or ORDER BY clause in connection with selecting a join order from among a plurality of join orders. Conventionally, the presence of a GROUP BY or ORDER BY clause could require an optimizer to be locked in to a particular join order, and claim 27, like claim 13, addresses this problem through the use of transitive closure analysis, and none of the references cited by the Examiner disclose or suggest the application of transitive closure analysis to a GROUP BY or ORDER BY clause for this recited purpose. Accordingly, claim 27 is patentable for many of the same reasons as set forth above for claim 13, and reversal of the rejection of claim 27 is respectfully requested.

Independent Claim 29

Claim 29 generally recites an apparatus including in part program code configured to apply transitive closure analysis to at least one search condition in a query that includes criteria that references a plurality of tables in order to re-order a result set generated for the query, and based on the transitive closure analysis, rewrite the criteria to generate modified criteria to reduce the number of tables referenced thereby. The criteria is one of a GROUP BY clause and an ORDER BY clause, and the program code is configured to apply transitive closure analysis to identify from the at least one search condition an equivalent field for a field referenced in the criteria, and to rewrite the criteria by substituting the equivalent field for the field referenced in the criteria.

As discussed above in connection with claim 1, none of the references cited by the Examiner disclose or suggest applying transitive closure analysis to a GROUP BY or ORDER

BY clause for the purpose of reducing the number of tables referenced thereby. In addition, there is a fundamental difference between reducing the number of rows referenced by a query and reducing the number of tables, so the Examiner's attempt to find analogy between the two is improper and is inconsistent with the understanding of those of skill in the art. Accordingly, claim 29 is patentable for many of the same reasons as set forth above for claim 1, and reversal of the rejection of claim 29 is respectfully requested.

Dependent Claims 3-5

Claims 3-5 are not argued separately.

Dependent Claim 9

Claim 9 depends from claim 1, and additionally recites that the database query involves a plurality of join operations, and that the method further includes running the query according to a join order that is based on the modified criteria. As discussed above in connection with claim 13, none of the references cited by the Examiner disclose or suggest applying transitive closure analysis to a GROUP BY or ORDER BY clause in connection with selecting a join order from among a plurality of join orders. Conventionally, the presence of a GROUP BY or ORDER BY clause could require an optimizer to be locked in to a particular join order, and claim 9, like claim 13, addresses this problem through the use of transitive closure analysis, and none of the references cited by the Examiner disclose or suggest the application of transitive closure analysis to a GROUP BY or ORDER BY clause for this recited purpose. Accordingly, claim 9 is patentable for many of the same reasons as set forth above for claim 13, in addition to its dependency on claim 1. Reversal of the rejection of claim 9 is respectfully requested.

Dependent Claims 15 and 18-19

Claims 15 and 18-19 are not argued separately.

Dependent Claim 26

Claim 26 depends from claim 25, and additionally recites running the query according to a join order that is based on the modified criteria. As discussed above in connection with claim 13, none of the references cited by the Examiner disclose or suggest applying transitive closure

analysis to a GROUP BY or ORDER BY clause in connection with selecting a join order from among a plurality of join orders. Conventionally, the presence of a GROUP BY or ORDER BY clause could require an optimizer to be locked in to a particular join order, and claim 26, like claim 13, addresses this problem through the use of transitive closure analysis, and none of the references cited by the Examiner disclose or suggest the application of transitive closure analysis to a GROUP BY or ORDER BY clause for this recited purpose. Accordingly, claim 26 is patentable for many of the same reasons as set forth above for claim 13, in addition to its dependency on claim 25. Reversal of the rejection of claim 26 is respectfully requested.

Dependent Claim 28

Claim 28 is not argued separately.

B. Claims 7-8 and 16-17 are patentable over *Pham*, *Ghazal*, and *Data Engineering*, and in further view of Chaudhuri

The Examiner has rejected claims 7-8 and 16-17 under 35 U.S.C. § 103(a) as being unpatentable over *Pham* in view of *Ghazal* in view of *Data Engineering* and further in view of U.S. Patent No. 5,598,559 to Chaudhuri (*Chaudhuri*). *Chaudhuri* is directed to an optimization technique for a query having a GROUP BY clause. The optimization technique generates execution plans, which places the GROUP BY preceding every internal join node. The optimizer then estimates the cost of each of these execution plans and selects the plan having the lowest estimated cost.

Dependent Claims 7-8

With respect to dependent claims 7 and 8, the Examiner contends that *Chaudhuri* discloses building indices over columns at col. 7, line 55 – col. 8, line 26. In this passage, *Chaudhuri* discloses a relation index which is used to optimize the sub-queries within a query containing at least one join. This passage fails to disclose building an index over a column. Furthermore, there is no disclosure in *Chaudhuri* to remedy the deficiencies of *Pham*, *Ghazal*, and *Data Engineering* identified above with respect to the rejection of independent claim 1, from which these claims depend. For these reasons, Applicant respectfully requests that the rejections of dependent claims 7 and 8 be reversed.

Dependent Claims 16-17

With respect to dependent claims 16 and 17, these claims depend from independent claim 13. As set forth above, *Chaudhuri* fails to remedy the deficiencies of *Pham*, *Ghazal*, and *Data Engineering*, and therefore for the same or similar reasons set forth above, Applicants submit that dependent claims 16 and 17 are also patentable. Furthermore, claim 16 refers to identifying a modified criteria that references a single, respective table for which an index exists, while claim 17 refers to identifying a modified criteria that references a single, respective table for which an index is to be created. Applicant submits that neither of these features, in combination with selecting a join order after applying transitive closure analysis to a GROUP BY or ORDER BY clause, is disclosed or suggested by any of the references cited by the Examiner. Applicant therefore respectfully submits that the rejections for these claims should be reversed.

CONCLUSION

Applicant respectfully requests that the Board reverse the Examiner's rejections of claims 1, 3-5, 7-9, 10, 13, 15-19 and 25-29, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at 513/241-2324. If any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

September 28, 2009
Date

/Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323
WOOD, HERRON & EVANS, L.L.P.
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
Telephone: (513) 241-2324
Facsimile: (513) 241-6234

VIII. CLAIMS APPENDIX: CLAIMS ON APPEAL (S/N 10/754,011)

Listing of Claims:

1. (Previously Presented) A method for optimizing a database query in a computer of the type including a database management system, the database query including criteria that references a plurality of tables in order to re-order a result set generated for the database query, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause, the method comprising the steps of, in the computer:

applying transitive closure analysis to at least one search condition in the query to identify an equivalent field for a field referenced in the criteria; and

based on the transitive closure analysis, rewriting the criteria to generate modified criteria to reduce the number of tables referenced thereby by substituting the equivalent field for the field referenced in the criteria, including rewriting the criteria to generate modified criteria that references only one table, based on the transitive closure analysis.

2. (Canceled).

3. (Previously Presented) The method according to claim 1, further comprising the step of:

determining if the criteria references a first field from a first table and a second field from a second table.

4. (Previously Presented) The method according to claim 3, wherein the rewriting step comprises the step of:

rewriting the criteria to reference the first field and a third field from the first table, wherein a first search condition in the query searches on a match between the first field and the second field, and a second search condition in the query searches on a match between the second field and the third field, and wherein applying transitive closure

analysis includes determining that the third field is equivalent to the second field in the criteria.

5. (Original) The method according to claim 1, further comprising the step of:

determining if the criteria references a plurality of tables.

6. (Canceled).

7. (Previously Presented) The method of claim 1, further comprising the step of:

building an index over a column of the one table.

8. (Previously Presented) The method of claim 1, further comprising the step of:

building an index over more than one column of a table among the plurality of tables.

9. (Original) The method according to claim 1, wherein the database query involves a plurality of join operations and the method further comprises the step of:

running the query according to a join order that is based on the modified criteria.

10. (Previously Presented) A method of optimizing a database query in a computer of the type including a database management system, the database query including criteria that operates to re-order a result set of the database query and requires creating a temporary file during operation, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause, the method comprising the steps of, in the computer:

applying transitive closure analysis to at least one search condition in the query to identify an equivalent field for a field referenced in the criteria; and

rewriting the criteria, based on the transitive closure analysis, to generate a modified criteria by substituting the equivalent field for the field referenced in the criteria, wherein the criteria references a plurality of tables and the modified criteria references a single table; said modified criteria operating to re-order a result set of the database query and avoid creating a temporary file during operation.

11.-12. (Canceled).

13. (Previously Presented) A method for optimizing a database query in a computer of the type including a database management system, the database query involving a plurality of join operations and a plurality of search conditions, the method comprising the steps of, in the computer:

applying transitive closure analysis to the plurality of search conditions in the query to determine a subset of equivalent search fields;

rewriting a criteria, that operates to re-order a result set of the database query, to generate a set of respective modified criteria that each reference one or more equivalent search fields, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause; and

selecting a join order from among a plurality of join orders for the plurality of join operations, including analyzing join orders using at least one of the set of respective modified criteria.

14. (Canceled).

15. (Previously Presented) The method according to claim 13, further comprising the step of:

running the query according to a join order, the join order determined by selecting one of the set of respective modified criteria.

16. (Original) The method according to claim 13, further comprising the step of:

identifying a subset of the respective modified criteria that reference a single, respective table and for which an index to that table exists.

17. (Original) The method according to claim 13, further comprising the step of:

identifying a subset of the respective modified criteria that reference a single, respective table and for which an index is to be created.

18. (Previously Presented) The method according to claim 17, further comprising the step of:

running the query according to a join order, the join order determined by selecting one of the subset of respective modified criteria.

19. (Original) The method according to claim 13, further comprising the steps of:

performing cost analysis on each of the set of respective modified criteria; and

running the query according to a join order, the join order determined based on the cost analysis.

20.-24. (Canceled).

25. (Previously Presented) A program product, comprising:

program code configured upon execution thereof to:

apply transitive closure analysis to at least one search condition in a query that includes criteria that references a plurality of tables in order to re-order a result set generated for the query, and based on the transitive closure analysis, rewrite the criteria to generate modified criteria to reduce the number of tables referenced thereby, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause, wherein the program code is configured to apply transitive closure analysis to identify from the at least one search condition an equivalent field for a field referenced in the criteria, and wherein the program code is configured to rewrite the criteria by substituting the equivalent field for the field referenced in the criteria; and

a recordable computer readable medium storing the program code.

26. (Previously Presented) The program product of claim 24, wherein the program code is further configured to:

run the query according to a join order that is based on the modified criteria.

27. (Previously Presented) A program product, comprising:

program code configured upon execution to:

apply transitive closure analysis to a plurality of search conditions to determine a subset of equivalent search fields within a database query involving a plurality of join operations and the plurality of search conditions, rewrite a criteria, that operates to re-order a result set of the database query, to generate a set of respective modified criteria that each reference one or more equivalent search fields, and select a join order from among a plurality of join orders for the plurality of join operations by analyzing join orders using at least one of the set of respective modified criteria, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause; and

a recordable computer readable medium storing the program code.

28. (Previously Presented) The program product of claim 22 further configured to:

run the database query according to a join order, the join ordered determined by selecting one of the set of respective modified criteria.

29. (Previously Presented) An apparatus, comprising:

at least one processor;

a memory coupled with the at least one processor; and

a program code residing in memory and executed by the at least one processor, the program code configured to apply transitive closure analysis to at least one search condition in a query that includes criteria that references a plurality of tables in order to re-order a result set generated for the query, and based on the transitive closure analysis, rewrite the criteria to generate modified criteria to reduce the number of tables referenced thereby, wherein the criteria is one of a GROUP BY clause and an ORDER BY clause, wherein the program code is configured to apply transitive closure analysis to identify from the at least one search condition an equivalent field for a field referenced in the criteria, and wherein the program code is configured to rewrite the criteria by substituting the equivalent field for the field referenced in the criteria.

IX. EVIDENCE APPENDIX

[APPLICATION NO. 10/754,011]

None.

X. RELATED PROCEEDINGS APPENDIX

[APPLICATION NO. 10/754,011]

None.